REMARKS

Upon entry of this paper, claims 1-6 and 8-20 are pending, of which claims 1, 9, and 16 are independent. No claims are amended. Applicant respectfully submits that the pending claims define over the art of record.

Objection to the Specification and Drawings

Applicant amends the specification to address the usage of the trademark SOLARIS and to add the reference character "70" to the relevant portion of the specification. Applicant respectfully submits that objection to the specification and drawings have been overcome in view of the amendments. Applicant respectfully requests that the Examiner reconsiders and withdraws the objection to the specification and drawings.

Claim Rejection of Claims 1-6 and 8-20 under 35 U.S.C. §102

Claims 1-6 and 8-20 are rejected under 35 U.S.C. §102(a) as being anticipated by Jato API (hereinafter JATO) disclosed by Andy Krumel in "Jato: The new kid on the open source block, Part 1" (March 2001; herein after JATO_One), "Jato: The new kid on the open source block, Part 2" (April 2001; hereinafter JATO_Two), and "Jato: The new kid on the open source block, Part 3" (May 2001; hereinafter JATO_Three). Applicant respectfully submits that JATO does not disclose each and every element of the pending independent claims. Specifically, JATO does not disclose the elements of providing a base object class that includes at least one method for converting between object-oriented programming language objects and XML data objects, wherein said at least one method determines the steps needed to perform the conversion and performs the conversion and for a given instance of the selected sub-class of the base object class, invoking the method to perform a conversion on the given instance.

The pending independent claims require a base object class including at least one method for converting between object-oriented programming language objects and XML data objects, and the at least one method determines the steps needed to perform the conversion and performs the conversion. In response to the previous arguments filed on February 18, 2005, the Examiner argues that the processing of the script (and thus, the translation) is accomplished by the org.jato.JavaToXml class and the SimpleJavatoXml helper class on page 3 of the current Office

Action. Hence, the Examiner agrees that the script defines the operation needed to translate a Java object to XML. Referring to the code in Listing 2 on page 4-5 in JATO_One, the Jato script written in XML is stored as a static String variable in the SimpleJavaToXml helper class and the script is not a method or stored in a method. *See* also page 6 line 2. Furthermore, the Examiner agrees that the Jato script is run by the org.jato.JavaToXml class with the SimpleJavaToXml helper class. Therefore, the static String storing the Jato script is supplied by the SimpleJavaToXml helper class so that the org.jato.JavaToXml class can run the Jato script from the command line to convert JAVA objects to XML. Therefore, JATO does not disclose a based class including a method that determines the steps needed to perform the conversion and performs the conversion as required by the pending independent claims 1, 9, and 16.

Furthermore, the pending independent claims require that for a given instance of a sub-class of the base object class, the at least one method that determines the steps needed to perform the conversion is invoked to perform a conversion on the given instance. In other words, an instance of a class can convert itself from a JAVA object to XML by invoking a method that the class inherent from its based class. However, JATO does not disclose this. Referring to section "The Java-toXML Jato script" on page 6 of JATO_One, four steps are listed to explain how the Jato script works. These steps show that the Jato script asks the Jato interpreter to get objects from the SimpleJavaToXml helper class and perform conversion for each objects obtained. However, these objects that are converted are not instances of either the SimpleJavaToXml helper class nor the org.jato.JavaToXml class. See code in Listing 2 on page 5 and first paragraph on page 7. Additionally, on top of page 8 shows the following code for translating Java objects to XML within an application:

```
JavaToXml jtox = new JavaToXml();
jtox.setrootOption("system-properties");
jtox.setSaveOption(true, "simple.xml");
jtox.setHelperClass("SimpleJavaToXml");

Document doc = jtox.transform();
```

An instance `jtox` of JavaToXml is created and a function called `transform()` is invoked on this instance. However, the `jtox.transform()` function does not convert the instance `jtox` from a Java object to XML, but rather converts application's objects to XML, which is saved to a filed named "simple.xml". See top portion of page 8 in JATA_One. Applicant respectfully submits that JATO does not disclose the element that for a given instance of the selected sub-class of a based object class, invoking a method inherent from the based class to perform the conversion on the given instance required by the pending independent claims 1 and 9 and dependent claims 17-18.

Dependent claims 2-6, 8, 10-15, 17-20 depend on one of the independent claims and include the patentable subject matter in the corresponding independent claim. Accordingly, Applicant respectfully requests that the Examiner reconsiders and withdraws the rejection of claims 1-6, and 8-20.

In view of the above amendment, Applicant believes the pending application is in condition for allowance.

Applicant believes no fee is due with this statement. However, if a fee is due, please charge our Deposit Account No. 12-0080, under Order No. SMQ-066 from which the undersigned is authorized to draw.

Dated: June 27, 2005

Respectfully submitted,

By Kevin J. C
Kevin J. Canning
Registration No.: 35,470
LAHIVE & COCKFIELD, LLP
28 State Street
Boston, Massachusetts 02109
(617) 227-7400
(617) 742-4214 (Fax)
Attorney For Applicant